**NIST Big Data Public Working Group (NBD-PWG)**

**NBD-PWD-2015/Use-case-#4-HC-with-Mahout**

**Source: NBD-PWG**

**Status: Draft**

**Title: Possible Big Data Use Cases Implementation using NBDRA**

**Author: Russell Reinsch (CFGIO), Afzal Godil (NIST), Wo Chang (NIST)**

To support Version 2 development, this unique Big Data use case (with publicly available datasets and analytic algorithms) for implementation is analyzed using the NIST Big Data Reference Architecture (NBDRA) as a backdrop. We encourage NBD-PWG members to help implement the use case using NBDRA so that we can learn about the dataflow as well as their interactions between NBDRA key components.

**Big data Analytics for Healthcare Data/Health informatics**

**Introduction**

Big data is defined as high-volume, high-velocity, and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision-making.

As large amounts of healthcare data are produced continually and stored in different databases, healthcare data certainly fits the definition of big data. It is estimated that in the US healthcare spending approximately, $75B to $265B is lost each year to healthcare fraud.1

**Algorithms:**

Develop statistical analysis, visualization, and machine learning tools to statistically analyze and develop predictive models for healthcare payment data and possibly detect irregularities and prevent healthcare payment fraud.

### Dataset: The Healthcare dataset:

### The Center for Medicare and Medicaid Services (CMS) (http://www.cms.gov), released in the dataset into the public domain known as “Medicare Part-B in 2014”. The dataset includes a set of records documenting about transactions between over 900,000 medical providers and CMS.

### Datasets can be found at: [Note: This Compressed ZIP package contains the tab delimited data file (Medicare\_Provider\_Util\_Payment\_PUF\_CY2012.txt) which is 1.7GB uncompressed and contains more than 9 million records, thus importing this file into Microsoft Excel will result in an incomplete loading of data.

### Use of database or statistical software is required; a SAS® read-in statement is supplied. Additionally, this ZIP package contains the following supporting documents: CMS\_AMA\_CPT\_license\_agreement.pdf, and [Medicare-Provider-Util-Payment-PUF-SAS-Infile.sas](https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Downloads/Medicare-Physician-and-Other-Supplier-PUF-SAS-Infile.zip)]

<http://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Physician-and-Other-Supplier2012.html>

Or the direct link to the dataset (446MB compressed; 1.7 GB after uncompressed) is:

<http://download.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Downloads/Medicare_Provider_Util_Payment_PUF_CY2012_update.zip?agree=yes&next=Accept>

### Possible Development Tools

**Big Data:**

Apache Hadoop, Apache Spark, Apache HBase, Apache Mahout, Apache Lucene/Solr, MLlib -Machine Learning Library

**Visualization:**

D3 Visualization, Tableau visualization

**Languages:**

Java, Python, Scala, Javascript, JQuery

Specific Questions:

What machine learning tools can be used for detecting irregularities in healthcare data?

**Use Case Sections:**

Primary actors and roles page 3

Level, preconditions and minimal guarantee page 3

Background on the tools implemented in the use case page 3

**Main flow scenario using Mahout page 3**

Subflows: visualization tools, model evaluation page 5

Extensions (dealing with errors) Follow each main flow scenario

Alternate flows page 5

Licensing and TCO page 5

Legend:

Activities in this use case that match the BDRA M0437 Activities View diagram are noted in red type.

Functions in this use case that match the BDRA M0437 Functional Components View diagram are highlighted in yellow and noted in red type.

Primary roles and actors

1. Analysts should have previous experience with Hadoop and machine learning (ML) technologies; as well as Java.
2. An Eligible ProfessionalE is incorporating medication information, laboratory diagnostics, and reconciling drug interaction checks and problem information. As the clinician at the point of care, the EP represents the prime source for medication reconciliation (NIST NCPDP Compatibility). The EP needs a consolidated medication list. Only C32 CCD “strictly requires coded drug identifiers...”
3. Nurses will undertake some of the same activities as the eligible professional; the nurse is also retrieving information such as imaging results. The nurse works with ICD 10 codes.

Level, preconditions and minimal guarantee

These use cases represent the first steps in modeling of processes and as such still require iteration cycles to reach accuracy and completeness. This use case states a generic analytical lifecycle, we suggest that project managers should seek advice from professionals with backgrounds specific to the Public Health domain.

Previous installation of Hadoop and Mahout software is required. Additional technologies including Cygwin (if implementing on a Windows based operating system), Maven (if installing with CDH distribution), Eclipse or Spark may be required.

Background on the OS anomaly detection technology implemented in this use case

Mahout is a Java library that implements supervised and unsupervised ML technologies for classification, clustering and recommendation. Mahout supports two kinds of k-means analysis, one for small data and “Streaming” for large. Streaming activities are not applicable in this use case.

**Main flow scenario: Mining with Random Forest in Mahout**

Certain fundamental tasks exist in most data mining projects.

1. Acquiring the data.
2. Identifying features in the data and decide on algorithms.
3. Ingest
4. Preparing data. Depending on which algorithm is determined to be most suitable, an analyst may take different approaches to preparing the raw data. In pre-processing / conversion: cleansing and transformation into appropriate format may be necessary prior to processing, to provide needed structure.
5. Run.
6. Validate results.

Acquiring the data

The analyst starts the project in their IDE of choice, and sets Mahout on the classpath. The data must be formatted into a “Datamodel” object by creating a CSV text file with each line referring to a user ID, item ID and score / or value triple (userID,itemID,value). The triple is referred to as a “Preference” in the Mahout Datamodel abstraction. This activity matches with the Application Provider container, Collection oval in the BDRAAV diagram.

If storing in HDFS, the dataset can be transferred to HDFS with a few short commands.1

This activity matches with the Framework Provider container, Infrastructure polygon, Store oval in the BDRAAV diagram. This function matches the Framework Provider container, Infrastructure polygon, Storage box; and Distributed File Systems box in the Platforms polygon of the BDRAFCV diagram.

Perform descriptive visualization

In this step the analyst gets a feel for the features in the dataset and identifies interesting variables or contributors. This function matches the Application Provider container, Visualizations box of the BDRAFCV diagram.

Integration and ingest:

Transformation: A descriptor file can be generated to label data in the training set so the algorithm can identify which set is categorical (and which is numerical).3 This activity matches with the Application Provider container, Preparation oval in the BDRAAV diagram. This function matches with the Application Provider container, Transformations box in the BDRAFCV diagram. Data management: Hcatalog for metadata.

Alternate flow: potential requirement to integrate with MUMPS and Intersystems Cache’. MUMPS was developed in the 1980s. It forms the basis of data management in the VA system and performs orchestrator range integration for Epic, the DOD, and other massive government offices. Health IT departments and data consumers have been building connectors and add-ons for transforming and interfacing with MUMPS ever since. As a core program, MUMPS works very well however there is a real demand amongst stakeholders to come up with something else, to modernize, as MUMPS is a very old system; however converting to a new vehicle in mid-flight is far from trivial.

Intersystem Cache’ is a hybrid relational / non-relational transaction engine, originally an OODBMS. Version 2 of this document will detail integration scenarios involving MUMPS and Intersystems Cache’.

The next step in the main flow scenario is to describe the jar, which requires three parameters, the path for the data, a descriptor file location, and data attribute information.

The triple is organized in a “UserPreferenceArray” and the Datamodel interface handles the loading of the data into Mahout.2

Processing:

The next task is to build the algorithm. Example code is presented in the cited link (1). Additional arguments that can be added with main class build forest commands include variable selection and number of trees to be grown / implemented. The new dataset will be classified and distributed by Hadoop. This activity matches with the Framework Provider container, Platforms polygon, Interactive Processing oval in the BDRAAV diagram. This function matches with the Application Provider container, Algorithms box in the BDRAFCV diagram.

Alternate flow: If MySQL is used for storage, access via JDBC.

Run:

Analysis is run in a confusion matrix and sent to a prediction folder, where the anomaly and normal dataset are defined in an output file. This function matches with the Application Provider container, Analytics oval in the BDRAAV diagram.

Visualization Subflow

TBD

Model evaluation

TBD

Licensing and TCO

Mahout is licensed Apache 2.0

Reference Information

1. White SE. Predictive modeling 101. How CMS’s newest fraud prevention tool works and what it means for providers. J AHIMA. 2011;82(9): 46–47.
2. Gupta. <https://books.google.com/books?id=UC_WBgAAQBAJ&pg=PA69&lpg=PA69&dq=mahout+anomaly+detection&source=bl&ots=Cz6R-N0HbU&sig=i8cOgOGFKukXE2pHZa4TvBpmOvE&hl=en&sa=X&ved=0CFYQ6AEwCTgKahUKEwiPy7iPnqTIAhXFWT4KHfH4BbA#v=onepage&q=mahout%20anomaly%20detection&f=true>
3. Transformation may not be required for outlier detection.
4. Mahout: <http://datascience.stackexchange.com/questions/2293/scalable-outlier-anomaly-detection>
5. List of roles in NIST.IR.7804-1